A variability-tolerant feedback technique for throughput maximization of TRBGs with predefined entropy

Tommaso Addabbo, Massimo Alioto, Ada Fort,
S.Rocchi, Valerio Vignoli

Department of Information Engineering
University of Siena, 53100 Italy
(e-mail: addabbo@dii.unisi.it)

⋆ RESEARCH MANUSCRIPT ⋆

– PLEASE REFER TO THE PUBLISHED PAPER[1] –

Abstract

In this paper a probabilistic feedback technique to maximize the throughput of a generic True Random Bit Generator (TRBG) circuit, under a given constraint on the entropy, is discussed. In the proposed solution, the throughput of the device is dynamically and adaptively varied by an on-line entropy detector, such to obtain, with an arbitrary confidence level, an entropy greater than a given worst-case value. The approach, which has a general validity, introduces a method for making maximum use of the TRBG random bit generation capabilities, maximizing the generation throughput while preserving its entropy. Differently from the classical 'open loop' TRBG design approach, in which the circuit parameter variability determines an uncertainty about the actual entropy of the device, with the proposed techniques the TRBG generation speed is varied under a given constraint on the entropy. The method can be applied to all those integrated TRBG circuits proposed in the literature and based on the uniform sampling of, e.g., random physical processes or chaotic dynamical systems.

1 Introduction

Random number generation represents a key task in several applications like cryptography [1], stochastic computation [2], testing of digital complex systems [3] and Monte Carlo simulations [4]. In these applications, randomness is used in *probabilistic* (or *randomized*) *algorithms*, i.e., in procedures in which chance is part of the logical flow [5]. In practice, this means that a machine implementing such an algorithm has access to a physical source of randomness, a *Random*

[1] Journal of Circuits, Systems and Computers, 2010, vol. 19, n. 4, p. 1-17.

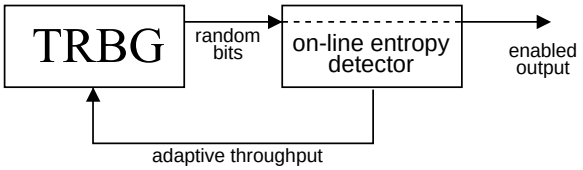


Figure 1: The block diagram of a "smart" TRBG.

Number Generator, for making stochastic choices during its execution, or to initialize registers with random values. A sequence of random numbers can be obtained using a sequence of bits generated by a Truly Random Bit Generator (TRBG).

TRBGs are devices usually implemented by mixed analog-digital circuits. In general they are based on the sampling of *intrinsically random physical processes* derived from, e.g., the electronic noise (Johnson noise, Schottky noise or some other stochastic phenomena present in silicon devices [6–8]), the unstable oscillations of high-jitter oscillators [9, 10], and the dynamics of chaotic systems [11–13]. A TRBG can be modeled as a *binary ergodic information source with positive entropy*, and its aim is to *approximate* the classical ergodic model defined for an unpredictable fair coin thrower: the greater is the entropy of the TRBG, and the better this approximation is achieved. Accordingly, the entropy is one of the indicators of the statistical quality of a TRBG.

Unfortunately, the behavior of a TRBG is usually highly sensitive to the variability and perturbation of circuit parameters caused, e.g., by the tolerances of the circuit fabrication process [14, 15], or induced by the environment housing the device (supply voltage oscillations, electromagnetic coupling, heating, etc.) [6–8, 11]. More in detail, the variability of the parameters typically introduces *bias* and *correlation* between the generated symbols, and a decrease of the TRBG entropy value. This problem can be countered either by improving the robustness of the circuit with respect to the parameter variability[2], or by lowering the throughput (in terms of issued random bits per second), in order to counteract the effects of correlation (by under-sampling or adopting suitable information compression algorithms).

In this paper we discuss a probabilistic feedback technique aiming at maximizing the throughput of a generic TRBG circuit, under a given constraint on the entropy. As a result, referring to Fig. 1, the throughput of the device is dynamically and adaptively varied on the basis of the on-line entropy estimation provided by a processing digital block, such to obtain an entropy greater than a given worst-case value with an arbitrary confidence level. The discussed approach has a general validity and *it can be adapted to any of the several output post-processing techniques proposed in literature in order to vary the generator entropy* (e.g., methods based on under-sampling or information compression) [8, 16–18].

Our approach introduces a method for making maximum use of the TRBG random bit generation capabilities determined also by the circuit implementation (process/voltage/temperature variations), dynamically maximizing the

[2]This issue usually leads to a significantly increased design effort, due to additive constraints, and an area/throughput/power penalty.

throughput given a certain worst case entropy constrain. This approach is different from the classical 'open loop' TRBG design approach, in which the circuit parameter variability determines an uncertainty about the actual entropy of the device. The method can be applied to all those integrated TRBG circuits proposed in the literature and based on the uniform sampling of, e.g., random physical processes or chaotic dynamical systems.

Note that in this paper the randomness test point of view for the validation of random binary source is not used. Randomness tests aim at verifying if a source is 'random enough' for a given application by checking if the ratio of finite sequences that pass a predefined statistical test suite is above a certain acceptance threshold [19–21]. In this paper we propose the idea of 'closed loop' TRBG design, providing a simple scalable low-complex method to counteract the circuit parameter variability. Of course, depending on the considered target application, a proper post-processing unit or proper sophisticated validation algorithms may be still required before issuing the generated random sequences to the final user circuits[3].

This paper is organized as in the following. In Section 2 we introduce some basic definitions about the Average Shannon Entropy (ASE) and the problem of its estimation. In Sections 3 and 4 we provide the main theoretical results on which the proposed entropy detector is based on, discussing in detail how practically the ASE can be estimated, even by means of a design example. In Section 5 we show the feasibility of the proposed technique by applying the method to a chaos-based TRBG. Finally, the conclusions and references close the paper.

2 The entropy and its estimation

In real TRBGs *bias* and *correlation* between symbols deteriorate the statistical characterization of the generated sequences, and Information Theory provides theoretical tools for measuring how well an ergodic binary source approximates the behavior of an unpredictable fair coin thrower.

Definition 1 *The* Average Shannon Entropy *(ASE) of a TRBG is equal to*

$$ASE = \lim_{k\to\infty} -\frac{1}{k} \sum_{\beta\in\{0,1\}^k} P(\beta)\cdot \log_2 P(\beta), \qquad \textit{[bit/time-step]} \tag{1}$$

where the summation extends to the finite set collecting the 2^k *binary* k*-tuples of the form* $\beta = \{b_0,\ldots,b_{k-1}\}, b_i \in \{0,1\}$*. In (1)* $P(\beta)$ *represents the generation probability for the* k*-tuple* β*, and it is assumed that if* $P(\beta) = 0$ *then* $P(\beta)\log_2 P(\beta)$ *is equal to zero. Since we are using logarithms with base* 2*, the result is expressed in bit/time-step.*

The ASE indicates, for a given TRBG, the average amount of information issued at each time-step. From the above definition it is immediate to check that for the ideal TRBG[4] the ASE is equal to 1 bit/time-step, i.e., it is equal to the maximum classical Shannon Entropy for a binary source. The computation

[3] E.g., in some cryptographic applications a specific method to detect the device failure or tampering.

[4] The classical ergodic model used for a fair coin thrower.

of (1) is unfeasible for most TRBGs, since the statistical distributions of any order are involved. Accordingly, the practical approach consists in truncating the expression (1) to a given integer $k > 0$. In such case we have the Partial Average Shannon Entropy[5]

$$\mathrm{ASE}_k = -\frac{1}{k} \sum_{\beta \in \{0,1\}^k} P(\beta) \cdot \log_2 P(\beta), \qquad \text{[bit/time-step]}. \tag{2}$$

The truncated expression (2) will be used in the following, in order to simplify the analysis.

2.1 Estimation of binary symbol generation probabilities

For any integer $k > 0$, the estimation of (2) is based on the direct estimation of the terms $P(\beta)$, with $\beta \in \{0,1\}^k$. Since we are assuming the binary source to be ergodic, it results that

$$\lim_{N \to \infty} \frac{\#\beta}{N} = P(\beta), \tag{3}$$

where $\#\beta$ is the number of occurrences for the symbol β among N observations. Accordingly, we define the estimation of $P(\beta)$ as $\tilde{P}_\beta = \frac{\#\beta}{N}$, that can be simply obtained counting the occurrences of the binary symbol β among N observations.

The random variable $N\tilde{P}_\beta$, that is obtained by counting the occurrences of β among N observations, is distributed over $\{0, 1, \ldots, N\}$ according to a probability mass function that can be, in general, extremely complicated to express in analytical form, even provided to know the exact statistical characterization of the ergodic binary source. To a first approximation, if we assume the generated symbols as a sequence of independent and identically distributed (i.i.d.) random variables, the probability for $N\tilde{P}_\beta$ to be equal to p is given by the well known binomial probability mass function

$$P(N\tilde{P}_\beta = p) = \frac{N!}{p!(N-p)!} P(\beta)^p \left(1 - P(\beta)\right)^{N-p}, \tag{4}$$

with mean value $NP(\beta)$ and variance $NP(\beta)(1 - P(\beta))$. For large[6] N, the previous expression can be approximated by a normal distribution with mean value $NP(\beta)$ and standard deviation

$$\sigma = \sqrt{NP(\beta)(1 - P(\beta))}. \tag{5}$$

Within this theoretical framework, the mean value of the estimation $\tilde{P}_\beta$ is exactly $P(\beta)$, and the accuracy of the estimation can be increased by lowering the standard deviation of $\tilde{P}_\beta$, i.e., by increasing the number of observations N. Moreover, referring to the 3σ confidence interval for Gaussian distributions, we recall that the variable $\tilde{P}_\beta$ is within $\pm 3\frac{\sigma}{N}$ away from its mean value $P(\beta)$ with probability $\approx 99.7\%$. The authors verified through computer simulations of binary Markovian sources with finite memory that the above theoretical Gaussian

[5] Of course, $\lim_{k \to \infty} \mathrm{ASE}_k = \mathrm{ASE}$.

[6] A rule of thumb says that N is large enough if both $NP(\beta)$ and $N(1 - P(\beta))$ are greater than 10.

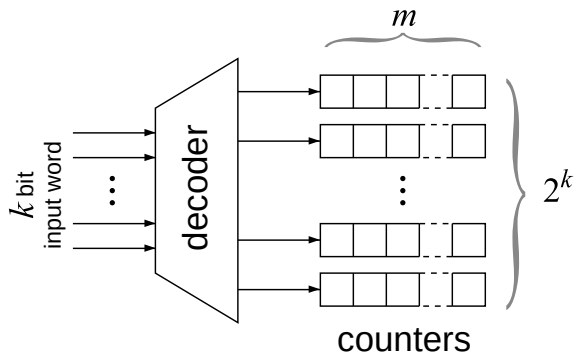


Figure 2: The schematic structure of the circuit for counting the occurrences of any k-bit symbol $\beta \in \{0,1\}^k$.

model is reliable when N is much larger than the memory of the stochastic process. In general, for an arbitrary ergodic source, it suffices that the first order correlation between symbols vanishes within a time-period reasonably short with respect to the width of the observation window. In the following, we assume the Gaussian approximation to be valid.

The 2^k terms $N\tilde{P}_\beta$ referred to the binary symbols $\beta \in \{0,1\}^k$ can be computed, e.g., using 2^k counters that count for each symbol the occurrences among N observations. A possible way to perform this task is by means of the circuit depicted in Fig. 2. Assuming to split a kN-bit long random binary sequence into N disjoint k-bit symbols, for each symbol a simple decoder enables the increment of the corresponding counter, as depicted in Fig. 2.

As far as the complexity of the circuit depicted in Fig. 2 is addressed, it must be highlighted that it increases proportionally to 2^k: this issue represents a limit to the implementation of this structure for large values of k (e.g., $k > 8$). In principle, the modulo m of each counter should be large enough to count up to N occurrences. In such case $m \approx \log_2 N$, and the total area required for the counters can be roughly estimated to be $2^k \log_2 N$ times the area of one flip-flop. Actually, m can be chosen much smaller than $\log_2 N$, reducing the circuit complexity, as discussed in the next sections.

3 Worst case partial entropy

For the aim of this work, we do not need to estimate the actual value of the TRBG entropy, but it is sufficient to estimate whether it is *above or below a given value*. Unfortunately, the definition (2) involves complex computations (logarithms and products), and it is therefore particularly expensive (in terms of computational cost) to be evaluated. In this section we discuss a simple method for determining a sufficient condition that guarantees that a binary source has an entropy greater than a given level, apart from an estimation error related to the terms $\tilde{P}_\beta$ which is taken into account in Section 4. It is worth noting that this condition is not necessary, i.e., the decision rule may discard sources with an acceptable entropy. On the other hand, the method requires

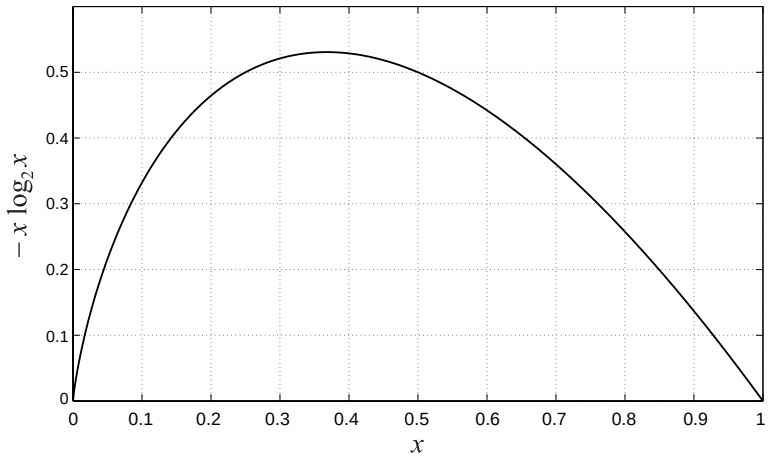


Figure 3: The function $-x\log_2 x$ and its first order approximation in the point x_0.

low-complex calculations[7], and hence it allows for the definition of a simple circuit implementation of an entropy detector with arbitrary confidence level, and it does not fail when it detects low-entropy sources: this is a critical issue when dealing, e.g., with cryptographic applications.

The summation in (2) is made of terms of the form $h(x) = -x\log_2 x$ (see Fig.3). It results that $\forall x_0 \in (0,1)$ the function h can be written as

$$h(x) = h(x_0) + h'(x_0)(x - x_0) - \xi(x - x_0), \tag{6}$$

where $-\xi(x - x_0)$ is equal to the sum of all the higher powers of the Taylor polynomial expression of the function h.

On this basis, for each positive integer k, and for any x_0, the partial entropy ASE_k can be written as

$$\mathrm{ASE}_k = -\frac{1}{k}\sum_{\beta\in\{0,1\}^k} P(\beta)\cdot\log_2 P(\beta) = \frac{1}{k}\sum_{\beta\in\{0,1\}^k} h(P(\beta)) = \tag{7}$$

$$= \frac{1}{k}\sum_{\beta\in\{0,1\}^k} [h(x_0) + h'(x_0)(P(\beta) - x_0) - \xi(P(\beta) - x_0)] = \tag{8}$$

$$= \frac{1}{k}\left[2^k h(x_0) + h'(x_0)(1 - 2^k x_0) - \sum_{\beta\in\{0,1\}^k} \xi(P(\beta) - x_0)\right]. \tag{9}$$

Since the previous expression holds for any $x_0 \in (0,1)$, we can chose x_0 equal to the value $P(\beta)$ for an ideal TRBG, i.e., $x_0 = \frac{1}{2^k}$, obtaining $h(x_0) = \frac{k}{2^k}$ and $(1 - 2^k x_0) = 0$. Therefore we can write

$$\mathrm{ASE}_k = \frac{1}{k}\left[k - \sum_{\beta\in\{0,1\}^k} \xi\left(P(\beta) - \frac{1}{2^k}\right)\right]. \tag{10}$$

[7] Also from a hardware implementation point of view.

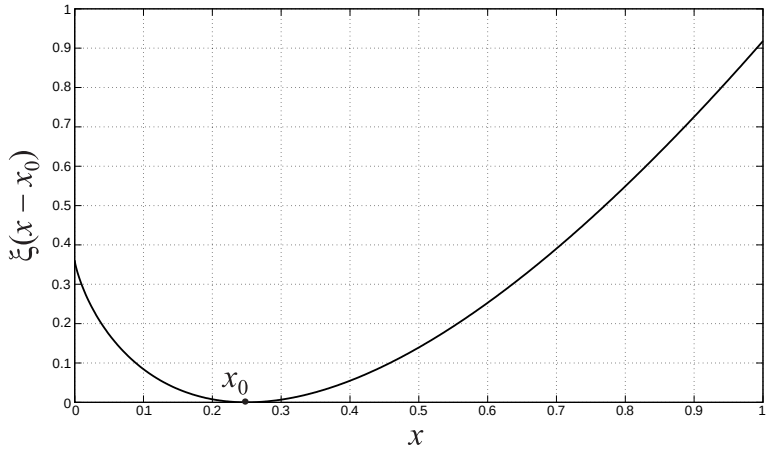


Figure 4: The error function $\xi(x - x_0)$ defined in (11), assuming $x_0 = \frac{1}{4}$.

It is interesting noting that the quantity $P(\beta) - \frac{1}{2^k}$ is equal to the distance between the actual probability $P(\beta)$ for the binary symbol β to be generated, and its value in an ideal TRBG.

Let us now focus on the terms $\xi\left(P(\beta) - \frac{1}{2^k}\right)$. From eq. (6) we have

$$\xi(x - x_0) = h'(x_0)(x - x_0) - h(x) + h(x_0), \tag{11}$$

which is 0 for $x = x_0$, and

$$\frac{d\xi}{dx}(x - x_0) = h'(x_0) - h'(x) = \frac{(\ln x + 1) - (\ln x_0 + 1)}{\ln 2} = \frac{\ln x - \ln x_0}{\ln 2}. \tag{12}$$

Accordingly, the second derivative

$$\frac{d^2\xi}{dx^2}(x - x_0) = \frac{1}{x \ln 2} \tag{13}$$

is positive $\forall x \in (0, 1)$. Concluding, the error function $\xi(x - x_0)$ is a non-negative convex function with minimum in x_0 (see Fig. 4).

We can now prove the following

Proposition 1 *Let the 2^k terms $P(\beta)$ in (2) belong to the interval $[\frac{1}{2^k} - \delta, \frac{1}{2^k} + \delta]$, with $0 < \delta < \frac{1}{2^k}$, i.e., by defining*

$$P_L = \frac{1}{2^k} - \delta \qquad \textit{and} \qquad P_H = \frac{1}{2^k} + \delta, \tag{14}$$

we assume

$$P_L \leq P(\beta) \leq P_H, \qquad \beta \in \{0, 1\}^k. \tag{15}$$

Accordingly, the minimum obtainable partial entropy is

$$\mathrm{ASE}_k|_{wc} = -\frac{2^k}{2k} \left[P_L \log_2 P_L + P_H \log_2 P_H\right]. \tag{16}$$

Proof. We first note that $\frac{2^k}{2}P_L + \frac{2^k}{2}P_H = 1$, i.e., a possible distribution of generation probabilities is that one for which one half of the 2^k binary symbols in $\{0,1\}^k$ has generation probability P_L, whereas the other half has generation probability P_H. In such case, it is trivial to verify from (2) that $\mathrm{ASE}_k = -\frac{2^k}{2k}\left[P_L \log_2 P_L + P_H \log_2 P_H\right]$. To conclude the proof, it suffices to show that for any other distribution of the 2^k terms $P(\beta)$ in the interval $[P_L, P_H]$, the partial entropy is not lower than (16). Accordingly, let us note that from (10) the worst case partial entropy (16) can be written as

$$\mathrm{ASE}_k|_{wc} = \frac{1}{k}\left[k - \left(\frac{2^k}{2}\xi_L + \frac{2^k}{2}\xi_H\right)\right], \tag{17}$$

where $\xi_L = \xi(P_L - \frac{1}{2^k})$ and $\xi_H = \xi(P_H - \frac{1}{2^k})$.

We recall that since the function $\xi(x - \frac{1}{2^k})$ is convex then for any x_1, x_2 it results $\forall t \in [0,1]$

$$\xi\left(x_2 - t(x_2 - x_1) - \frac{1}{2^k}\right) \leq t\xi\left(x_1 - \frac{1}{2^k}\right) + (1-t)\xi\left(x_2 - \frac{1}{2^k}\right). \tag{18}$$

Since x_1 and x_2 are arbitrary values, we can set in the above inequality $x_1 = P_L$ and $x_2 = P_H$, and for any point y between P_L and P_H a value $t_y \in [0,1]$ exists such that $y = P_H - t_y(P_H - P_L)$. We now note that from the normalization property of probability mass functions it follows that $\sum_{\beta\in\{0,1\}^k} P(\beta) = 1$, which implies

$$\sum_{i=1}^{2^k}(P_H - t_i(P_H - P_L)) = \sum_{i=1}^{2^k}\left(\frac{1}{2^k} + \delta - 2\delta t_i\right) = 1, \tag{19}$$

and, after simple calculations,

$$\sum_{i=1}^{2^k} t_i = \sum_{i=1}^{2^k}(1 - t_i) = \frac{2^k}{2}. \tag{20}$$

On the other hand, from the convexity of $\xi(x - \frac{1}{2^k})$ we have

$$\sum_{\beta\in\{0,1\}^k}\xi\left(P(\beta) - \frac{1}{2^k}\right) = \sum_{i=1}^{2^k}\xi\left(P_H - t_i(P_H - P_L) - \frac{1}{2^k}\right) \leq \tag{21}$$

$$\leq \sum_{i=1}^{2^k}(t_i\xi_L + (1-t_i)\xi_H) = \xi_L\sum_{i=1}^{2^k} t_i + \xi_H\sum_{i=1}^{2^k}(1-t_i). \tag{22}$$

Summarizing, the previous inequality and eq. (20) imply that for any distribution of the 2^k terms $P(\beta)$ in the interval $[P_L, P_H]$

$$\sum_{\beta\in\{0,1\}^k}\xi\left(P(\beta) - \frac{1}{2^k}\right) \leq \frac{2^k}{2}\xi_L + \frac{2^k}{2}\xi_H. \tag{23}$$

Accordingly, the partial entropy can not be lower than the worst case (17), concluding the proof.

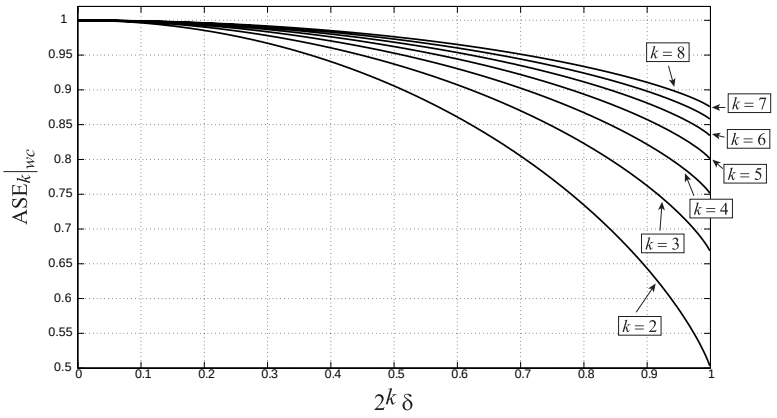


Figure 5: The worst case partial entropy as a function of the maximum allowed normalized deviation $2^k\delta$ for the terms $P(\beta)$ in eq. (2), for different values of k.

According to the previous result, the Fig. 5 reports the worst-case $\mathrm{ASE}_k|_{wc}$ (16) as a function of the maximum allowed normalized deviation $2^k\delta$ for the terms $P(\beta)$ in eq. (2), taking into account different values for k. In this figure, when the normalized deviation $2^k\delta \to 1$, the worst case partial entropy is obtained referring to the distribution of generation probabilities $P(\beta)$ such that one half of the 2^k binary symbols in $\{0,1\}^k$ has generation probability $P_L \to 0$, whereas the other half has generation probability $P_H \to \frac{2}{2^k}$, i.e., twice the ideal generation probability $\frac{1}{2^k}$.

On the basis of the eq. (16), or on the basis of Fig. 5, a precise maximum acceptable deviation δ can be identified for the terms $P(\beta)$ in eq. (2) such that the partial entropy is guaranteed to be greater than a given worst case value. For example, by choosing $k = 5$ and $\delta = \frac{0.5}{2^k} = \frac{1}{64}$, if all the generation probabilities $P(\beta)$ belong to the interval $[\frac{1}{64}, \frac{3}{64}]$, the worst case partial entropy $\mathrm{ASE}_5|_{wc}$ is guaranteed to be greater than 0.96 bit/time-step.

4 The generation probability estimator

The results in the previous section can be used provided to know whether the generation probabilities $P(\beta)$ lie within $P_L = \frac{1}{2^k} - \delta$ and $P_H = \frac{1}{2^k} + \delta$ for all the binary symbols $\beta \in \{0,1\}^k$. Under the assumptions introduced in subsection 2.1, in this section we discuss the correct design of the generation symbol probability estimator.

From (5) we know that the the estimation of $P(\beta)$ has standard deviation

$$\sigma_\beta = \sqrt{\frac{P(\beta)(1-P(\beta))}{N}}. \tag{24}$$

Accordingly, for any given N, the variance of $\tilde{P}_\beta$ is a simple quadratic form with maximum in $P(\beta) = 0.5$. Therefore, if we restrict our analysis to those $P(\beta)$ values such that $P_L \le P(\beta) \le P_H$, assuming $k > 1$ and $0 < \delta < \frac{1}{2^k}$ the worst-case standard deviation is obtained for $P(\beta) = P_H = \frac{1}{2^k} + \delta \le \frac{1}{2}$.

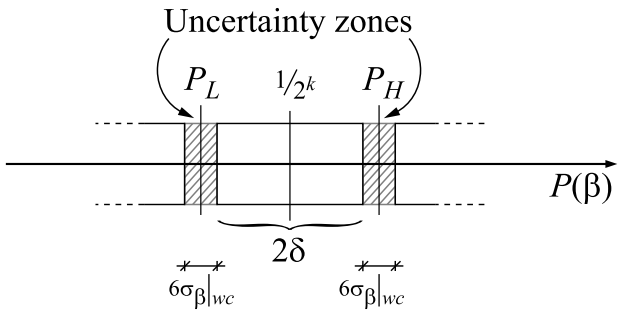


Figure 6: The two uncertainty zones around P_H and P_L containing those values $P(\beta)$ which may cause a wrong evaluation about the actual entropy of the TRBG.

Accordingly, by varying N, we can set the worst-case standard deviation of $\tilde{P}_\beta$ lower than any arbitrary value ρ, obtaining the inequality

$$\sigma_\beta|_{wc} = \sqrt{\frac{P_H(1-P_H)}{N}} < \rho. \tag{25}$$

The previous inequality is satisfied if

$$N > \frac{P_H(1-P_H)}{\rho^2}. \tag{26}$$

For each binary symbol $\beta \in \{0,1\}$ the estimation of $P(\beta)$ is therefore affected by an error $\tilde{P}_\beta - P(\beta)$ which may cause the entropy detector to make an error. In detail, for each binary symbol $\beta \in \{0,1\}$, the error is made if

$$P(\beta) \in (P_L, P_H) \qquad \text{and} \qquad \tilde{P}_\beta \notin (P_L, P_H), \tag{27}$$

or if

$$P(\beta) \notin (P_L, P_H) \qquad \text{and} \qquad \tilde{P}_\beta \in (P_L, P_H). \tag{28}$$

The analytical calculation of the previous probabilities can be carried out only making strong assumptions on the statistical characterization of the TRBG to be evaluated. Nevertheless, the set of $P(\beta)$ values which can be accepted is not hard-limited by P_L and P_H, since certain values $P(\beta)$ greater than P_H or lower than P_L can be accepted as good values due to the estimation error. Since we assumed $\tilde{P}_\beta$ to be a Gaussian random variable, we can identify two *uncertainty zones* around P_H and P_L and wide about $6\sigma_\beta|_{wc}$, containing those values which may cause a wrong evaluation of the actual entropy of the TRBG (see Fig. 6). Accordingly, assuming $2\delta > 3\sigma_\beta|_{wc}$, the $P(\beta)$ values greater than $P_H + 3\sigma_\beta|_{wc}$ (or values lower than $P_L - 3\sigma_\beta|_{wc}$) have probability roughly lower than 0.3% to be estimated between P_L and P_H. Similarly, the $P(\beta)$ values between $(P_L + 3\sigma_\beta|_{wc}, P_H - 3\sigma_\beta|_{wc})$ have probability rougly lower than 0.3% to be estimated greater than P_H or lower than P_L. Taking into account these considerations and adopting the worst-case point of view introduced in the Proposition 1, the expression (16) can be corrected as

$$\mathrm{ASE}_k|_{wc} \approx -\frac{2^k}{2k}\left[P'_L \log_2 P'_L + P'_H \log_2 P'_H\right], \tag{29}$$

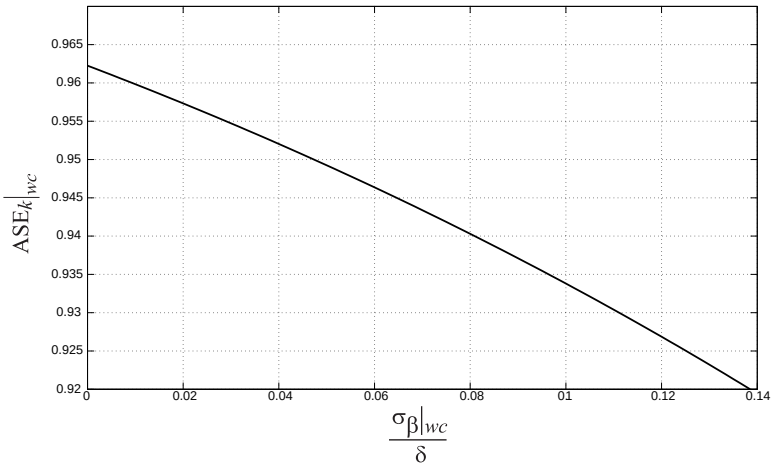


Figure 7: The worst case partial entropy (29) for the example discussed in subsection 4.1, taking into account the effects of the estimation errors.

where

$$P'_L = \frac{1}{2^k} - \delta - 3\sigma_\beta|_{wc} \qquad \text{and} \qquad P_H = \frac{1}{2^k} + \delta + 3\sigma_\beta|_{wc}. \tag{30}$$

On the basis of the previous results the generation probabilities estimator can be designed as discussed in the following example.

4.1 Design example

According to the sufficient condition (16), let us design a system for detecting whether the 5-bit partial entropy for a TRBG is lower than 0.96 bit/time-step, with a tolerance of ±0.01 bit/time-step. Thus, we are assuming $k = 5$ and, by inspecting Fig. 5, $\delta \approx \frac{0.5}{2^5} = \frac{1}{64}$, whereas the accepted values for the generation probabilities $P(\beta)$, according to (15), are bounded by $P_L = \frac{1}{64}$ and $P_H = \frac{3}{64}$. On the basis of (29) it is possible to plot Figure 7, according to which, in order to have a partial entropy $0.95 < \mathrm{ASE}_k|_{wc} < 0.97$, we set $\sigma_\beta|_{wc} < \rho \approx 0.048\delta = 0.75 \cdot 10^{-3}$. Finally, by means of (26), we obtain $N = 59600$.

4.1.1 Circuit implementation

The implementation of the device discussed in the previous example requires $2^5 = 32$ counters whose aim is to detect whether $NP_L < N\tilde{P}_\beta < NP_H$. For implementation simplicity, since $N = 59600$ is the minimum value for N such to satisfy the inequality (26), one can chose for N a value equal to the smallest power of 2 greater than 59600, i.e., $N = 2^{16} = 65536$. In such case, the aim of the 32 counters is to detect whether

$$2^{16}\frac{1}{64} = 1024 < 2^{16}\tilde{P}_\beta < 2^{16}\frac{3}{64} = 3072. \tag{31}$$

Accordingly, the previous condition can be verified for each of the 32 symbols using 12-bit counters (i.e., counting from 0 to 4095), assuming each counter

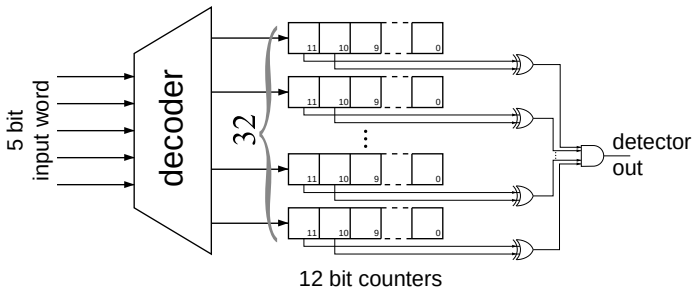


Figure 8: The schematic structure of the efficient circuit implementing the core of the system for detecting whether the 5-bit partial entropy for a TRBG is assured to be greater than a nominal worst case level of 0.96 bit/time-step, with a tolerance of ±0.01 bit/time-step (Subsection 4.1).

capable to stop the counting process when reaching the overflow. As a result, when the counting is finished, the condition (31) is verified if the most significant bits stored in the counters are equal to either '01...' or '10...'[8].

Referring to the schematic hardware structure described in Fig. 8, the working mechanism for the entropy detector can be summarized as in the following. At the start-up all the registers and counters are reset. The TRBG output is then split into $2^{16} = 65536$ non-overlapping 5-bit symbols, i.e., 2^{21} bits are examined 5 bits at a time. For each 5-bit symbol the decoder enables the corresponding counter, increasing its value by one unit. After completing the counting of 65536 symbols, the detector output is high if and only if the condition (29) is satisfied. On the basis of this signal, referring to fig. 1, the throughput of the TRBG can be increased or lowered.

In this example, the core hardware architecture of the entropy detector requires a 5-bit decoder, 32 12-bit counters , 32 2-inputs `xor` gates, a logic block computing the equivalent output of a 32-inputs `and` gate, a 16-bit counter for counting the 65536 symbols and few more logic for interfacing the input and the output signals with the TRBG.

4.2 Hardware implementation of a generic entropy detector

As highlighted in the previous example, the efficient implementation of the entropy detector can be achieved when, for two properly chosen integers $n, p > 0$, the parameter N is chosen equal to 2^n, and $2^k\delta$ is chosen equal to $\frac{1}{2^p}$. In such case, in order to obtain the partial entropy (16) the generic condition (15) implies that

$$2^{n-k} - \frac{2^{n-k}}{2^p} \leq N\tilde{P}_\beta \leq 2^{n-k} + \frac{2^{n-k}}{2^p}. \tag{32}$$

Accordingly, the size m of the counters can be chosen such that $2^m = 2 \cdot 2^{n-k}$, and the output of the detector can be easily obtained with few logic gates having as an input the most significant $p+1$ bits of the 2^k counters used for the counting.

[8]Indeed, in such case the final counting value range from 1024 to 3071.

For different values of N and δ, more complex circuits are required for verifying the condition $NP_L < N\tilde{P}_\beta < NP_H$ for each binary symbol (e.g., digital comparators). Nevertheless, if restricting the parameter space (N, δ, k) to the low-complex particular cases mentioned above, the different achievable possibilities satisfy most the needs in practical cases, and the circuit implementation (e.g., by a FPGA device) is straightforward.

5 Application to a real TRBG

Let us consider the chaos-based TRBG proposed in [11]. In this TRBG the unbiasing can be assumed negligible, whereas depending on the nominal parameter values of the chaotic circuit a typical correlation affects the statistical characterization of generated bits, as depicted in the upper-left subplot of Fig. 9. In this section we apply the entropy detector proposed in this paper to vary the TRBG throughput by means of the direct TRBG under-sampling[9]. In this example the entropy detector was implemented by an Altera FPGA Cyclone II EP2C5T144C6, and for the design we set a target worst-case 5-bit partial entropy equal to 0.99 bit/time-step, assuming a tolerance equal to ± 0.005 bit/time-step. According to the previous results and discussions, we set $k = 5$, $2^k\delta = \frac{1}{4}$ and $N = 131072$, obtaining $NP_H = 5120$ and $NP_L = 3072$. With these values the implementation of the entropy detector required 522 logic elements ($\approx 11\%$ of the logic elements available in the specific FPGA).

The Fig. 9 reports experimental data obtained from the hardware prototype described in [11], equipped with the entropy detector. As it can be seen in the upper subplots of this figure, for the considered TRBG the entropy detector detected a low-entropy event when assuming a $1:1$ under-sampling rate (indeed, in this case the actual 5-bit partial entropy was ≈ 0.9218 bit/time-step). By lowering the TRBG throughput, assuming a $2:1$ under-sampling rate, the detector detected again a low-entropy event. It is interesting noting that the actual 5-bit partial entropy was ≈ 0.9907 bit/time-step (greater than the requested worst-case value), but the sufficient condition (16) was not satisfied. Finally, adopting a $3:1$ under-sampling rate, the sufficient condition (16) was satisfied, obtaining an actual 5-bit partial entropy equal to ≈ 0.9994. As it can be seen in Fig. 9, by lowering the throughput of the TRBG, the 5-bit symbols are progressively better uniformly distributed, increasing the overall quality of the TRBG statistical characteristic.

6 Conclusions

In this paper we have discussed a probabilistic feedback technique to maximize the throughput of a generic TRBG circuit, under a given constraint on the entropy. In the proposed solution, the throughput of the device is dynamically and adaptively varied by an entropy detector, such to obtain, with an arbitrary confidence level, an entropy greater than a given worst-case value. The approach, that has a general validity, introduces a method for making maximum use of

[9] In this section the under-sampling method was assumed for simplicity of presentation and because the considered TRBG can be suitably modeled as an unbiased source with vanishing autocorrelation function (see [11] for further details). As discussed in the Introduction, several alternative solutions can be adopted.

the TRBG random bit generation capabilities, leading the generation throughput to its maximum while preserving its entropy. As discussed with examples, the method requires low-complexity circuits and it can be applied to all those integrated TRBG circuits proposed in literature and based, e.g., on the uniform sampling of random physical processes, or of chaotic dynamical systems.

References

[1] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997.

[2] F. Klebaner, *Introduction to Stochastic Calculus with Applications*. Imperial College Press, 2005.

[3] R. David, *Random Testing of Digital Circuits: Theory and Application*. Marcel Dekker, 1998.

[4] J. Gentle, *Random Numbers Generation and Monte Carlo Methods*, 2nd ed. Springer-Verlag, 2003.

[5] T. Addabbo, M. Alioto, A. Fort, A. Pasini, S. Rocchi, and V. Vignoli, "A class of maximum-period nonlinear congruential generators derived from the Rényi chaotic map," *IEEE Transactions on Circuits and Systems - I*, vol. 54, no. 4, pp. 816–828, 2007.

[6] R. Ohba, D. Matsushita, K. Muraoka, S. Yasuda, T. Tanamoto, K. Uchida, and S. Fujita, "Si nanocrystal MOSFET with silicon nitride tunnel insulator for high-rate random number generation," in *Proc. of IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, 2006.

[7] D. C. Ranasinghe, D. Lim, S. Devadas, D. Abbott, and P. H. Cole, "Random numbers from metastability and thermal noise," *IEEE Electronics Letters*, vol. 41, no. 16, pp. 13–14, 2005.

[8] C. Petrie and A. Connelly, "A noise-based IC random number generator for applications in cryptography," *IEEE Transaction on Circuits and Systems I*, vol. 47, no. 5, pp. 615–621, 2000.

[9] S. Petrie and A. Connelly, "Modeling and simulation of oscillator-based random number generators," in *Proc. ISCAS 96*, vol. 4, May 1996, p. 324327.

[10] M. Alioto, L. Fondelli, and S. Rocchi, "Analysis and performance evaluation of area-efficient true random bit generators on FPGAs," in *Proc. of ISCAS 2008, Seattle (USA)*, 2008, pp. 1572–1575.

[11] T. Addabbo, M. Alioto, A. Fort, S. Rocchi, and V. Vignoli, "A feedback strategy to improve the entropy of a chaos-based random bit generator," *IEEE Transaction on Circuits and Systems – part I*, vol. 53, no. 2, pp. 326–337, 2006.

[12] S. Callegari, R. Rovatti, and G. Setti, "Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos," *IEEE Trans. on Signal Processing*, vol. 53, no. 2, pp. 793–805, 2005.

[13] S. Ozoguz, A. Elwakil, and S. Ergun, "Cross coupled chaotic oscillators and application to random bit generation," *IEE Proc. Circuits, Devices & Systems*, vol. 153, no. 5, pp. 506–510, 2006.

[14] T. Addabbo, A. Fort, S. Rocchi, D. Papini, and V. Vignoli, "Invariant measures of tunable chaotic sources: Robustness analysis and efficient estimation," *IEEE Transactions on Circuits and Systems - I*, 2008, accepted.

[15] T. Addabbo, M. Alioto, A. Fort, S. Rocchi, and V. Vignoli, "Entropy enhancement in a chaos-based random bit generator," in *2006 International Symposium on Nonlinear Theory and its Applications (NOLTA'06), Bologna (Italy)*, 2006, pp. 375–378.

[16] Y. Peres, "Iterating Von Neumann's procedure for extracting random bits," *The Annals of Statistics*, vol. 20, no. 1, pp. 590–597, 1992.

[17] J. Von Neumann, "Various techniques used in connection with random digits," *Applied Mathematics Series, U.S. National Bureau of Standards*, vol. 12, pp. 36–38, 1951.

[18] M. Blum, "Independent unbiased coin flips from a correlated biased source: a finite state markov chain," in *25th Ann.Symp. on Foundations of Computer Science*, 1984, pp. 425–433.

[19] "NIST Special Publication 800-22: A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2001.

[20] S. Pincus and B. Singer, "Randomness and degrees of irregularity," *Proc. Natl. Acad. Sci. USA*, vol. 93, pp. 2083–2088, 1996.

[21] S. Pincus and R. Kalman, "Not all (possibly) "random" sequences are created equal," *Proc. Natl. Acad. Sci. USA*, vol. 94, pp. 3513–3518, 1997.

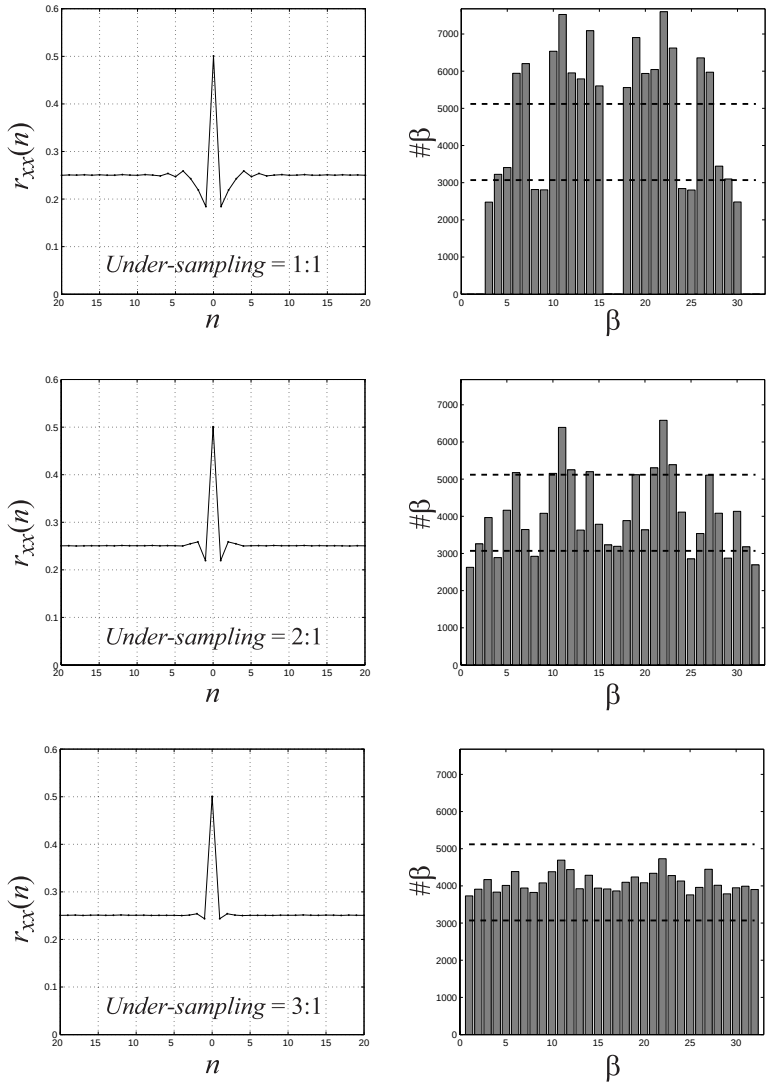


Figure 9: Results obtained applying the entropy detector to a chaos-based TRBG. By considering three different under-sampling rates, for each case the autocorrelation function between generated random bits (left column) and the experimental obtained histogram used to verify the condition (16) are reported.